Challenges for the Self-Safety in Autonomous Vehicles

Matthieu Carre, Ernesto Expósito, Javier Ibañez-Guzmán

Challenges for the Self-Safety in Autonomous Vehicles

Matthieu Carré
Univ Pau & Pays Adour, E2S UPPA, LIUPPA, EA3000
64600 Anglet, France
Email: matthieu.carre@renault.com

Ernesto Exposito
Univ Pau & Pays Adour, E2S UPPA, LIUPPA, EA3000
64600 Anglet, France
Email: ernesto.exposito@univ-pau.fr

Javier Ibanez-Guzman
Renault S.A.S,
1 av. du Golf, 78288 Guyancourt, France.
Email: javier.ibanez-guzman@renault.com

***Abstract*—The combination of multiple functions having different and complementary capabilities enables the emergence of Autonomous Vehicles. Their deployment is limited by the level of complexity they represent together with the challenges encountered in real environments with strong safety concerns. Thus a major concern prior to massive deployment is on how to ensure the safety of autonomous vehicles despite likely internal (e.g. malfunctions) and external (e.g. aggressive behaviors) disturbances they might undergo. This paper presents the challenges that undergoes the design and development of autonomous vehicles with respect to their functional architecture and adaptive behaviors from a safety perspective. For the purpose of the rationales, we define needs and requirements that lead to the formulation of an architectural framework. Our approach is based on paradigms and technologies from non-automotive domains to address non-functional system properties like safety, reliability and security. The notion of micro-services is also introduced for the self-safety of autonomous vehicles. These are part of the proposed framework that should facilitate the analysis, design, development and validation for the adequate composition and orchestration of services aimed to warrant the required non-functional properties, such as safety. In the present paper, we introduce the structural and behavioral adaptations of the framework to offer a holistic and scalable vision of the safety over the system.**

I. Introduction

Within the last ten years, road vehicles are becoming progressively more automated with the use of advanced electric and electronic (E/E) systems and driver assistance systems (ADAS). The current transition to enhanced automation in automotive industry is leading to new types of Autonomous Vehicles (AV). At this date, SAE International has provided enhanced recommendations for autonomous vehicles in their last report [1]. The levels 4-5 Automated Driving System-operated vehicles shall delegate all the driving tasks from the driver even the fallback procedures in case of failures. Humans driving errors [2] mostly emanate from the complexity to perceive, to understand and to act accordingly upon a complex and dynamic driving context. On the other hand, the large diversity of functions and configurations required for driving operations has largely limited the development and deployment of automated vehicle solutions. In particular, ensuring safety requires to characterize, to understand, to model and to manage both the safety of the driving tasks by taking the appropriate actions and the safe operation in case of failures. These perspectives show how the representation of safety and its integration can be very complex and challenging. In fact, some non-functional properties that need to be guaranteed (e.g. safety, reliability, security, performance) can only emerges in real-time operations. In addition, another source of complexity comes from the large diversity of external and internal actors involved in the system life-cycle alongside to the large panel of internal and external vehicle services that need to be offered. Those external and internal actors are referring to both humans (i.e. users, designers, experts and stakeholders in all the system life-cycle) and the system items (i.e. collaborative systems, sub-systems, components, functions, capabilities, etc.).

Autonomous systems for road vehicles have highly inherited from the project of Albus et al. [3] for military unmanned vehicle systems. Borrowing from cognitive psychology and neuroscience, the reference model architecture has brought structure and hierarchy to components and relationships. This work has introduced practical methods to deal with context with sets of layers of abstraction in the same functional architecture. On this basis, further works have proposed solutions to solve part of the complexity of the domain by identifying the degradation states for well-adapted decisions and actions [4, 5], apprehending the range and limitation of actions of the system [6, 7], evolving the architecture of the system to provide better adaptation behaviors, and finally, integrating non-functional dimensions [8].

However, with the advent of automated vehicles that requires more and more understanding of its context, the necessity to build an extendable, context-aware, evolvable and pluggable architecture has arisen at run-time [9]. In fact, levels 4-5 ADS-operated vehicles require a well-adapted architecture aimed at developing the appropriate functions, capabilities and skills to adapt to complex and dynamic environment situations while still guaranteeing non-functional properties, as safety in any situations [8, 10]. Besides, the surge of micro-services paradigm [11] and the Internet of Things technologies [12, 13] are providing new semantic and abstraction tools for the knowledge representation and architecture management for the field. Certainly, the adoption of service-oriented architecture will provide more scalability and evolvability while staying manageable for the industry[8]. For sure, some trade-off analysis need to consider these solutions in a real-

time constrained environment, and overview the respect of non-functional properties to ensure acceptable operation and effective adaptations to the context.

The present research is aimed at identifying and designing the basis for a framework based on micro-services for the self-safety in automated vehicles. This self-* capability involves both a behavioral adaption of the system upon safety and a structural adaptation upon the context of the vehicle in our approach. In the paper, we characterize non-functional dimensions - in especial the safety - to make them understandable at the different layers of the system for run-time assurance. In particular, the framework would adapt the composition of the system architecture based on the dimensions that need to be verified (e.g. safety), using its associated knowledge and the current context.

This paper firstly introduces in Section II the levels of automation for automated vehicles commonly accepted as standards. Moreover, Sections II-B and II-C will present the related high-level capabilities for the functional architecture, and how those architectures have progressively integrated the management of safety. Furthermore, in Section III we explore the adaptive or evolvable architectures for safety with in the literature and current trends. Then, we present guidelines for the design and development of an architectural framework of AV for safety assurance in Section IV. Finally, Section V presents our framework based on micro-services for the self-safety in AV.

II. Current context in Autonomous Vehicles

This section presents the 6 levels of autonomy proposed by the Society of Automotive Engineers (SAE) and 7 high-level capabilities required for driving. They illustrate the main sources of complexity and challenges for functional architecture of AV. Then, frameworks and concepts related with safety for automated vehicle are described.

A. Levels of automation

At this date, SAE International has revised their previous vision on levels of automation and has provided recommendations for automated driving [1]. Those levels of autonomy provide general guidelines to describe the capabilities that the automated system needs to offer, and determine how technologically advanced one vehicle can be. The report has paved the way to illustrate the required capabilities and needs in the multidisciplinary and interdisciplinary environment of automated vehicles, system engineering and robotics.

The SAE levels of automation recommend the attribution of the responsibility to the Driver and/or the System. The term System refers an intelligent system that allows the vehicle to purse automation of the Dynamic Driving Task (DDT). Those levels also characterize the capabilities and the compatible environment of the System.

In our domain of interest of levels 4 and 5, all DDT and fallback are operated by the System. Such operations involve the system to perform online self-diagnosis and self-adaptation without any human interaction. The Driver have also the possibility to interfere with the system and can request control back according the system usage.

B. Main high-level capabilities for functional architecture of the automated vehicle

Based on the literature available [3, 9, 14], the Automated Driving System has to operate 7 high-level operational capabilities. Each of these capabilities captures and consolidates operational needs in the functional architecture. In Table I, we list those capabilities and describe the main functions that they accomplish.

However, those capabilities and functions may need to act differently in operation depending of the complexity and the dynamicity of the environment. An axis sensitive to the context dealing with the complexity and uncertainty of the system needs to be integrated. For this purpose, those specific functional considerations need to be captured into separate alternative configurations, distinct profiles, or abstracted in some use cases. Furthermore, non-functional properties also need to be captured during conception from regulators, standards, policy of the company that can be quite heterogeneous and dependent to the context and functions.

C. Evolution of the management of safety as a non-functional dimension in vehicle architectures

Safety mainly refers to the absence of harm from the system to the user(s) and the environment (driver, passengers, road users, others road entities) [15]. Considering safety for ADS-operated vehicle results in examining the harm the system can cause to the environment (e.g. through malfunction, dysfunction, fault, capability limitation) or the environment can cause on the vehicle (e.g. with obstacles, aggressive behavior of other users, unexpected events, etc.). In our current approach, we will first consider the behavioral safety for safe driving. In addition, functional safety is also an objective as the ability to deliver trusted services (avoidance of non-acceptable failures).

Safety have been considered and managed differently in AV, especially in ISO26262 [15] and in future SOTIF [16]. The compliance to standards and research on more self-awareness and safety in AV have successively introduced innovations at different scale to the System:

1) Structure for reasoning and self-awareness with a hybrid architecture both deliberative and reactive
2) Degraded modes to monitor closer the state of the system and components
3) Limitations consciousness of the system at run-time based capabilities and skills performances of the functional architecture
4) Constructionist and evolvable architecture to manage scalability and complexity of the System

In the first place, the structure of the framework of Albus on 4D/RCS [3] introduces a hybrid control architecture being both deliberative (i.e. actions based on reasoning and planning) and reactive (i.e. fast actions based on direct and simple condition on feedback). On the one hand, the framework structure ensures reasoning and planning processes based on goals and

Capabilities	Functions and roles
Localization	Localize the vehicle, estimate pose and provide corrections
Map data	Access information regarding roads for planning and navigation
Connectivity systems like V2V, V2I, V2X, Cloud services	Exchange Information regarding road usage and states, scene occlusion, prediction of potential safety threats and other external data sources access
Environment perception	Perceive static and dynamic elements; their states, dynamics and behaviors are also considered
Mission planner	Take Strategic, Tactical and Operational navigation scale decision and generate behavior to accomplish the mission
Self-Perception (also called Self-Awareness or World Model)	Understand the scene of the road elements and the vehicle's current state (functional capabilities and abilities, motion, ...)
Operation	Command and Control vehicle's actuators

TABLE I
CAPABILITIES AND FUNCTIONS IN THE AUTONOMOUS VEHICLE TO PERFORM DYNAMIC DRIVING TASKS

priorities of the decision entities within a control hierarchy. On the other hand, reactive loops that provide a faster and controlled response are introduced at each level of the control hierarchy, and can locally modify planned actions to adapt to new events. This approach is close in its concept and structure composition to the autonomic computing paradigm [17] introduced by IBM, with a hierarchy of autonomic orchestrating managers in IT infrastructure.

In their work to define robust system architectures, Tas et al. [5] emphasizes the relation between architectural design and the use of degraded operation modes, introduced by [4]. The authors refer to the design of a functional and layered system architecture as one of the main focus of autonomous driving to deal with safety-critical challenges in systems engineering. Moreover, they suggest that the use of an effective monitoring system is necessary to give proper feedback to the vehicle about its state and to allow to take well-adapted decisions.

In the matter of monitoring, Reschka et al. [18] reports the needs to provide a permanent online monitoring of vehicle capabilities, as well an adequate modeling tool to support appropriate and safe decisions. Accordingly, the framework perceives the current performance of the system during operation, by knowing the range of actions of the system and its limitations. With respect to ISO26262 standard [15], the authors propose the use of ability and skill graphs to design the functional architecture of AV. Those skills reflect the performance feedback of the system and then are used for system self-perception.

Regarding the long-term evolution of AV in systems engineering, Behere et al. [8] overview the key functional components and orientations needed for autonomous driving, to establish a layered evolvable functional architecture. They report the necessity of constructivist architectures managed by Artificial Intelligence to tackle the limitations of current engineering practices for scaling to more complex systems. These so-called constructivist architectures introduce a fundamental shift from manually designed to self-organizing architectures that evolve at run-time. The current challenge resides in the possibility of run-time reasoning and run-time verification of the desired properties as safety constraints. The design and implementation of such architectures will involve paradigms and technologies from non-automotive domains. For example, the use of reflective intelligent control systems based on a high-level supervisor on the functional layer that will allow the monitoring and change of its behavior for better adaptations.

Those approaches have been introduced progressively in the automotive domain in order to propose solutions for well-adapted decisions; to apprehend the range and limitation of actions of the system; to evolve the structure to provide better adaptation behaviors; and finally, to integrate non-functional dimensions.

III. STATE OF THE ART OF RECONFIGURABLE AND ADAPTIVE SYSTEM ARCHITECTURES

In this section, we present several approaches that deals with similar research problem of assuring self-adaptation from the safety perspective. Also, we include in this analysis not only works that specifically target safety, but also ones that share similar concerns on other relevant non-functional dimension like performance. Our analysis is based on criteria reflecting the solutions used for integration and scalability, the architecture design for reconfiguration, and finally the representation of safety (see Table II) to that end. Those criteria have been selected based on the behaviors of the services identified in [19] for the support of self-configuration in an adaptive and reconfigurable service-oriented architecture.

The first criteria in Table II reflects the solution used in the approaches based on integration and scalability. Then, the second criteria of context adaptation describes the ways of management and understandability for the contextual information in real-time. The third criterion, called architecture reconfiguration, covers the mechanisms of adaptation to ensure the safety. For the safety as a non-functional dimension, the fourth criteria characterizes how the dimensions and parameters of the context are integrated. Finally, the fifth criteria details the knowledge bases associated to the safety to predict and avoid risky situations.

Based on Table II, most of the reconfigurable architectures propose either verification and validation of the model at run-time, to provide support actions [18] or either adapt this model and reconfigure the resources to fit to the new context [22, 26]. Also, the solutions don't have the same scale for integration and range of actions. [25] propose a whole system whereas the solution in [28] acts as external to an existing system . In

References	Solution type for integration and scalability	Context adaptation	Architecture reconfiguration	Specific aspect covered	Knowledge bases type and data
[20, 21]	Fail-operational E/E Architecture for Highly-automated Driving Functions	Detects handle and error in the decentralized and independent network	Drives by fail-operational behavior driven of the key safety-critical functions	Fail-safe requirements of individual functions	Uses validated data generated at design to ensure consistency on individual decisions
[22, 23]	Management of different sensor setups using Model@run-time [24]	Monitors health at run-time for components generic function	Performs dynamic resources reconfiguration based on adaptive graceful degradation on functions	Analyzes satisfaction of the system safety properties and assess functions acceptability	Constructs a domain-specific model for data dependencies between functions and properties
[6, 18, 25]	Surveillance and safety system based on performance criteria and functional degradation	Monitors system and establish performance criteria	Takes functional degradation actions based on performance criteria to ensure a safe state	Identifies performance criteria through relevant parameters and their impact	Performance criteria for autonomous driving influencing vehicle control and maneuvers
[26]	Scalable Model-based autonomic context management system (ACoMS)	Provides fault tolerant provisioning of context information	Can dynamically configure and reconfigure its context information gathering and pre-processing functionality	Discovers and use new context information sources from features	Context models capture the relevant concepts and relations required by context-aware applications system-wide
[27]	Extensive architectural contract-based framework for components and functionalities integration and reconfiguration	Enables appropriate composition of modeled components and functionalities by constraint satisfaction	Search and adapts with the contracts the system configuration of components to a valid and complying state	Uses a central constraint solver on the problems of each respective component views and try to obtain a solution incrementally	Contracts for the components: properties required and to be guarantee
[28]	Agent-based decision support for run-time assurances by validation of monitored decisions	Delegates part of the decision-making to agents	Expresses positive feedback into reward for improve future decisions improvement	Detects safety-critical invariant violation in monitored agents and prevent execution, the agent is then tamed	Safety-critical invariant and goals are modeled and contextualized by the World Model component
[29]	Reconfiguration matrix as a data structure that correlates the chosen service	Selects or composes the best service to perform a specific mission	Provides an automated deployment machinery based on the micro-services characterization	Discover new services at run-time with the auto-description of the micro-service	Semantic description of the micro-service needs and properties

TABLE II
COMPARISON BETWEEN RECONFIGURABLE SYSTEM ARCHITECTURES UPON CONTEXT CHANGES.

this external approach, Mallozzi et al. [28] delegates part of decision-making process to agents, and consequently shift the assurance of the safety at run-time.

In their representation of non-functional dimension, Reschka et al. [6] propose to introduce relationships between concepts like ability, and skills to establish the monitoring of the system's operation and functional capabilities online. Thus, they provide a method to maintain or reach a a safe state of operation with safety adaptation actions [25].

We also observe the solutions that target scalability and flexibility in their integration tends to propose a service-oriented architecture with independent knowledge bases [20, 29].

From the collection of those works, we observe three trends emerging in the field of vehicle automation. The first trend would focus on solutions tend to be more distributed into components or services and form a hierarchy of decision process in the software architecture to break down the complexity of systems [20, 21, 28, 29]. This type of reflective structure allows monitoring and changing its behavior for better adaptations at different levels of the architecture. The second trend represents the use of constructivist architectures managed by AI to tackle the limitations of current engineering practices by integrating heterogeneous processes at different scales of the system [22, 23, 27]. The third trend acknowledges the usage of knowledge bases separated from the execution processes, and allow bringing semantic and inference in the architecture, at conception or at run-time [18, 26, 29].

Based on this analysis, the next section will introduce our rationales to propose an alternative approach regarding safety in its modeling and assurance. As a matter of fact, we will reuse some considerations of the presented methods to complete the approach to define, achieve, prove functional safety and safe operation during conception and at run-time.

IV. RATIONALES FOR A SAFETY-ORIENTED REFERENCE ARCHITECTURE

The state of the art proposes different solutions that permit to generate safe adaptation actions for the context. However, most of the reviewed published literature on adaptive systems for autonomous vehicles reveals focus on safety in particular perspectives and dependability in general but without much focus on a comprehensive and holistic vision of safety. For example, the solutions are either too specific to fit our approach on safety or either they produce a tight-coupled systems. For some solutions close to our perspectives, we determine that having a service-oriented architecture would solve difficulties in scaling system ability.

For this purpose, we want a scalable solution to provide a comprehensive assurance for safety by performing adaptations to guarantee this dimension according to the context. It consequently requires the involvement of heterogeneous sources of knowledge describing the dimension: high level goals, policies, indicators, regulators, and the future standards in the domain. In addition, the framework needs to be possibly integrated to existing functional architectures as additional layer. Therefore, the system must have the ability to understand what the dimension is; perceive, measure and analyze the safety;

and provide decision support to the functional architecture or directly perform reactive actions.

Our proposition addresses the problem of functional safety allowing monitoring and adaptation to context changes and validation at run-time. Based on existing and recent recommendations in the domain [9], a such solution would require the following elements:

- Decomposition of the adaptation mechanisms (domain logic) in diagnosis and adaptation processes for the safety and for the context
- Knowledge bases representing semantically the safety dimension, the context and the relation between them. Safety would be represented semantically as a non-functional dimension built from requirements, policies and standards. Part of the safety knowledge bases would describe the services capabilities (System services goals, functions, data flows and interfaces).
- Semantic description of the data-sources in the architecture
- Communication layer allowing semantic and scalable exchanges between the heterogeneous components and services of the different layers.

In the next section, we introduce and provide a first overview of the framework based on micro-services for the self-safety in AV.

V. A framework and methodology based on micro-services for the self-safety in Autonomous Vehicles

The framework proposes a scalable system providing diagnosis and adaptation capabilities to ensure safety upon the context. For these respectives high-level capabilities, we have identified 3 three layers:

1) Interface to the sensors and actuators: Provides an interface with the existing ADS functional system to access data and to send actions and orders;
2) Grid of micro-services (Behavioral adaption for safety): Ensures an acceptable level of safety by the performance of adaptation actions at run-time. The actions are produce by chains of independent functional processes and domain logic in safety;
3) Semantic Orchestrator (Structural adaptation for context): Orchestrates the grid of micro-services at run-time aims to optimize the safety coverage with respect to the context.

Figure 1 illustrates the respective interactions between the layers. We position the layers in the perspective of the requirements stated in Section IV and the previous capabilities. In the same section, we introduce the technological choices and their general purpose in the system, and finally the synergies in Table III.

A. First layer: Interface to the ADS functional system

The first layer acts as touchpoints managing the access to the managed resources (existing ADS, sensors and actuators) in the framework. This interface provides access to contextual information available in the managed resources and allows reactive actions to be send to the managed resources. It semantically describes the data and the component interfaces in order to deal with the heterogeneity of the sources. This layer also monitors the state and configuration of the managed resources.

B. Second layer: Grid of micro-services

The grid of M,A,P,E micro-services constitutes the second layer and performs the adaptation for safety with specialized micro-services. We have broken down the capabilities to perceive, understand, and mitigate safety according to the steps of the MAPE-K control loop [17] so that the capabilities that formed parts of a single process or indicators are isolated into composite and tiny micro-services. Each of the micro-services have now a category (Monitor, Analyze, Plan or Execute) and a respective knowledge base and goal in the system. The connections between those multiple micro-services result in a grid that evaluates safety through various indicators and performs mitigation.

C. Third layer: Semantic Orchestrator

The Semantic Orchestrator adapts the safety assurance upon the context by managing the structure of the grid network and the semantic descriptions of the composed micro-services. The third layer ensures that only relevant indicators and processes are running and correctly configured by orchestrating their deployment as micro-services. For this purpose, the Semantic Orchestrator performs the reconfiguration of the 2nd layer and thus adapt how the safety dimension is evaluated and mitigated upon the current context. Table III compiles the main interactions and impacts between the technologies choices and provides the role of the Semantic Orchestrator.

This MAPE-K adaptation process is also decomposed into stateful micro-services to perform the tasks of monitoring, analyzing, planning and execution. They are monitoring several aspects of the system to this end: context change, configuration of managed resources and safety performance of L2. The third layer can consequently determine if the factors raised by the symptoms (e.g. new context, new configuration and insufficient performance) requires an adaptation using inference. The inference process uses ontologies that describes the safety (indicators) and the context (internal and external) to provide a high-level understanding and to support inference at run-time. Then, we create a reconfiguration plan based on the list of relevant safety indicators to deploy that fit with the current available resources. Finally, an automated deployment solution reconfigures or instantiates the M, A, P, E micro-services and create appropriate bindings between the micro-services. Those bindings are driven by semantic needs described in the semantic description of the needs of each micro-service in the knowledge bases.

Each of these steps have a respective usage of the knowledge bases for inference, update of models and deployment using cookbook recipes. Firstly, the ontology that represents the

safety dimension in the framework describes the relevant indicators used in the 2nd layer and relates the specialization of individual entities. It is used in the Analyze and Plan steps of the third layer to evaluate the actual and future grid of M,A,P,E micro-services by inference and specializes the generic stateless components into a specific domain. Secondly, the context ontology presents the relations between the use cases, scenes and situations and the relevant indicators to use for the available data sources. The observable state of the environment and the configuration of the system (i.e. managed resources and second layer micro-services) are used in the Monitor and Analyze steps in the third layer to determine the need of a context adaptation. Finally, a cookbook stores the processes of adaptation and resources of the micro-service instances. In addition, a specialization of the knowledge is used to fixed the goals and policy of the M,A,P or E micro-service instances. Those knowledge bases are used by the automated deployment solution in the third layer to instantiate and specialize the instance of the second layer upon the context.

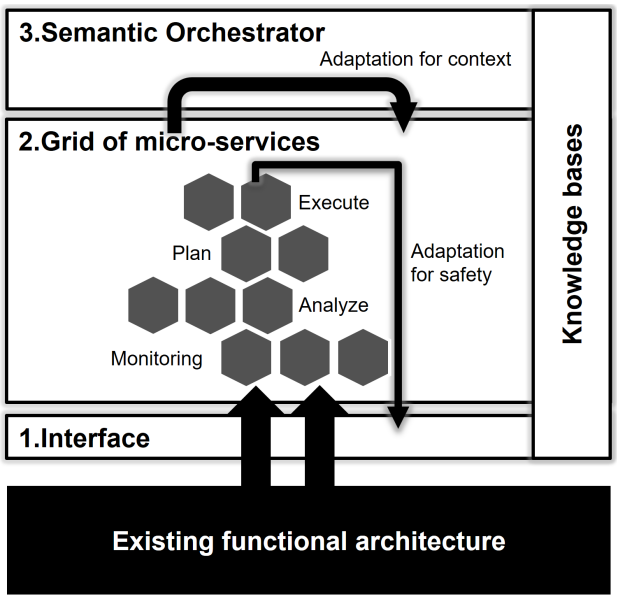

Fig. 1. Framework layered architecture

D. Knowledge Bases used in the framework

Ontologies are knowledge representations with well-defined and typed entities, properties and semantic relationships between those entities. These representations are flexible enough and can be easily shared and manageable for our intended usage. In our approach, we are using ontologies at different scales in the framework when semantic description, semantic relations or inference capability are required. Some ontologies capture the knowledge of the whole non-functional dimension and permit to allocate a relevant subset of the system to a local knowledge base. In this way, the system remains adaptable and scalable with an level of abstraction acceptable for reasoning and processing time. In this approach, we are using knowledge bases as follow:

- Static Knowledge Bases
  - Ontology for context: Represents the context of the Autonomous Vehicles (abstractions from use cases, scenes, situations);
  - Ontology for safety: Safety indicators with decomposition into MAPE, coverage and needs;
  - Semantic relations linking both ontologies to capture the requirements, abstractions and the goals of each indicators upon the different abstracted contexts;
  - Cookbook recipes: Contains the configurations and processes for each micro-service deployment.
- Models modified at run-time by the framework
  - Context model representing the observable state of the environment (internal and external) on Configuration of the managed resources and Configuration of the M,A,P,E micro-services grid;
  - Plan model containing the actions for the deployment.

VI. Conclusions, perspectives and future works

In this paper, we have introduced the current challenges for the design and development of autonomous vehicles with respect to their functional architecture and adaptive behaviors from a safety perspective. In respect of the current literature solutions, we have contributed to formulate a definition of needs and requirements for a such reference architecture based on paradigms and technologies from non-automotive domains that can address non-functional system properties like safety. In our present research, we propose a framework for run-time safety assurance and verification relevant to the context. This framework based on micro-services is composed of three layers, and proposes run-time reasoning and adaptation for the self-safety of AV.

The consensus view in the literature seems to integrate the consideration of safe adaptations and context management at run-time to palliate the level of complexity of AV deployment. However, research on a comprehensive and holistic vision of safety includes either too specific views on safety or propose tight-coupled solutions on standards. Compared to the existing solutions in the domain of safety, the framework has the scalability and evolvability to propose a holistic vision over the safety of the system upon different contexts. The automated deployment of different types of safety indicators provides the system different perspectives more than a simple aggregation and would result in an operational safety coverage for safety understanding. This capability requires high-level knowledge bases that describe both the dimension of safety, the possible context and relations between the concepts allocated in the indicators.

In the second layer of the framework, the safety assurance and validation are performed by adaptation mechanisms based on safety indicators. Those processes are decomposed into

	Autonomic Computing	Micro-services	Semantic Model (ontology)	Semantic Orchestrator
Autonomic Computing	-	Ensure orchestration of service and composition over the AM hierarchy	Split of MAPE steps to specific knowledge	Overview as a supervisor for all instances of M, A, P, E on a specific domain
Micro-services	Implement communication and service generic container	-	One functionality keep size small	Bare minimum of centralized orchestration of these services
Semantic Model (ontology)	Model the knowledge bases for MAPE steps	Semantic description of service needs, interfaces and information (Aspect-oriented programming)	-	Represent our interoperable knowledge and model the dimension

TABLE III
IMPLICATIONS OF THE TECHNOLOGIES COMBINATION AND IMPACT ON FRAMEWORK AND SEMANTIC ORCHESTRATOR (READ FROM ROW TO COLUMN)

specific M, A, P, or E micro-services instances. Their services capabilities (System services goals, functions, data flows and interfaces) are described semantically in the knowledge representation of the safety. As a result, this architecture style ensures scalable exchanges between heterogeneous managed resources and services in the system. In addition, it provides the possibility to be managed at a higher-level, and to be adapted to fit a new context where another vision of safety is perhaps required (e.g. from requirements, policies and standards). For this purpose, the Semantic Orchestrator can perform the reconfiguration of the micro-services composition using inference on the current, and the relations between safety indicators and the context. By holding a separation between the knowledge representation of the dimensions and the internal processes, the framework can easily integrate modifications and extensions regarding requirements, policies and future standards.

Future steps intend to establish a methodology to refine the architecture and the semantic description of components and the dimension of safety for a specific use case. We expect to create such knowledge representations over the context in the design and experiment with the framework on the ROS middleware. A future contribution will extend the presentation of our framework based on micro-services for the self-safety in AV and specify the methodology.

ACKNOWLEGMENT

We thank J. Aguilar for his support throughout this work. We also thank Renault-Nissan Alliance and UPPA for their financial and institutional support of the underlying research of this article.

REFERENCES

[1] S. O.-R. A. V. S. Committee *et al.*, "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems," SAE International, Tech. Rep., Sep. 2016, revision of J3016_201401.

[2] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey. (traffic safety facts crash stats. report no. dot hs 812 115)," Washington, DC: National Highway Traffic Safety Administration., techreport, Feb. 2015. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812115

[3] J. S. Albus, H.-M. Huang, E. R. Messina, K. Murphy, M. Juberts, A. Lacaze, S. B. Balakirsky, M. O. Shneier, T. H. Hong, H. A. Scott *et al.*, "4d/rcs version 2.0: A reference model architecture for unmanned vehicle systems," *NIST Interagency/Internal Report (NISTIR)-6910*, 2002.

[4] J. Lygeros, D. N. Godbole, and M. E. Broucke, "Design of an extended architecture for degraded modes of operation of ivhs," in *American Control Conference, Proceedings of the 1995*, vol. 5, Jun 1995, pp. 3592–3596 vol.5.

[5] Ö. Ş. Taş, F. Kuhnt, J. M. Zöllner, and C. Stiller, "Functional system architectures towards fully automated driving," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016, pp. 304–309.

[6] A. Reschka, G. Bagschik, S. Ulbrich, M. Nolte, and M. Maurer, "Ability and skill graphs for system modeling, online monitoring, and decision support for vehicle guidance systems," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, June 2015, pp. 933–939.

[7] S. Ulbrich, A. Reschka, J. Rieken, S. Ernst, G. Bagschik, F. Dierkes, M. Nolte, and M. Maurer, "Towards a functional system architecture for automated vehicles," *CoRR*, vol. abs/1703.08557, 2017. [Online]. Available: http://arxiv.org/abs/1703.08557

[8] S. Behere and M. Törngren, *Systems Engineering and Architecting for Intelligent Autonomous Systems*. Cham: Springer International Publishing, 2017, ch. 13, pp. 313–351.

[9] R. Johansson, J. Nilsson, C. Bergenhem, S. Behere, J. Tryggvesson, S. Ursing, A. Söderberg, M. Törngren, and F. Warg, *Functional Safety and Evolvable Architectures for Autonomy*. Cham: Springer International Publishing, 2017, pp. 547–560.

[10] R. Johansson, "Efficient identification of safety goals in the automotive e/e domain," in *8th European Congress on Embedded Real Time Software and Systems (ERTS 2016)*, Toulouse, France, Jan. 2016. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01292289

[11] M. Fowler and J. Lewis, "Microservices: a definition of this new architectural term," *ThoughtWorks. http://martinfowler.com/articles/microservices.html [last accessed on July 06, 2016]*, 2014. [Online]. Available: http://martinfowler.com/articles/microservices.html

[12] ——, "Consumer-driven contracts: A service evolution pattern," *ThoughtWorks. https://martinfowler.com/articles/consumerDrivenContracts.html [last accessed on July 27, 2016]*, 2016. [Online]. Avail-

able: http://martinfowler.com/articles/microservices.html

[13] M. Möstl, J. Schlatow, R. Ernst, H. Hoffmann, A. Merchant, and A. Shraer, "Self-aware systems for the internet-of-things," in *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, Oct 2016, pp. 1–9. [Online]. Available: http://ieeexplore.ieee.org/document/7750988/

[14] S. Behere and M. Törngren, "A functional reference architecture for autonomous driving," *Information and Software Technology*, vol. 73, pp. 136 – 150, 2016. [Online]. Available: https://sagar.se/files/wasa2015.pdf

[15] *ISO 26262 - Road vehicles – Functional safety*, International Organization for Standardization Norm ISO 26 262, 2011. [Online]. Available: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=43464

[16] *ISO/WD PAS 21448 - Road vehicles - Safety of the intended functionality. Under development.*, International Organization for Standardization Norm ISO/WD PAS 21 448, 2017. [Online]. Available: https://www.iso.org/standard/70939.html

[17] J. Kephart, D. Chess, C. Boutilier, R. Das, and W. E. Walsh, "An architectural blueprint for autonomic computing," *IBM White paper*, Jun. 2006. [Online]. Available: https://pdfs.semanticscholar.org/0e99/837d9b1e70bb35d516e32ecfc345cd30e795.pdf

[18] A. Reschka, J. R. Böhmer, T. Nothdurft, P. Hecker, B. Lichte, and M. Maurer, "A surveillance and safety system based on performance criteria and functional degradation for an autonomous vehicle," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012, pp. 237–242.

[19] R. Anthony, A. Rettberg, D. Chen, I. Jahnich, G. de Boer, and C. Ekelin, *Towards a Dynamically Reconfigurable Automotive Control System Architecture*. Boston, MA: Springer US, 2007, ch. 3, pp. 71–84.

[20] G. Weiss, P. Schleiss, C. Drabek, A. Ruiz, and A. Radermacher, *Safe Adaptation for Reliable and Energy-Efficient E/E Architectures*. Cham: Springer International Publishing, 2018, pp. 1–18.

[21] G. Weiß, P. Schleiß, and C. Drabek, "Fail-operational e/e architecture for highly-automated driving functions," *ATZelektronik worldwide*, vol. 11, no. 3, pp. 16–21, 2016.

[22] J. Frtunikj, V. Rupanov, A. Camek, C. Buckl, and A. Knoll, "A safety aware run-time environment for adaptive automotive control systems," *Embedded real-time software and systems (ERTS2)*, vol. 3, 2014. [Online]. Available: https://pdfs.semanticscholar.org/b1fc/44c745d9bd13d41a035f06d97b96524357d8.pdf

[23] J. Frtunikj, V. Rupanov, M. Armbruster, and A. Knoll, *Adaptive Error and Sensor Management for Autonomous Vehicles: Model-Based Approach and Run-Time System*. Cham: Springer International Publishing, 2014, pp. 166–180.

[24] U. Aßmann, S. Götz, J.-M. Jézéquel, B. Morin, and M. Trapp, *A Reference Architecture and Roadmap for Models@run.time Systems*. Cham: Springer International Publishing, 2014, pp. 1–18.

[25] A. Reschka, G. Bagschik, and M. Maurer, *Towards a System-Wide Functional Safety Concept for Automated Road Vehicles*. Cham: Springer International Publishing, 2018, pp. 123–145.

[26] P. Hu, J. Indulska, and R. Robinson, "An autonomic context management system for pervasive computing," in *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*, March 2008, pp. 213–223.

[27] J. Schlatow, M. Moestl, and R. Ernst, "An extensible autonomous reconfiguration framework for complex component-based embedded systems," in *Autonomic Computing (ICAC), 2015 IEEE International Conference on*, July 2015, pp. 239–242.

[28] P. Mallozzi, "Combining machine-learning with invariants assurance techniques for autonomous systems," in *Proceedings of the 39th International Conference on Software Engineering Companion*, ser. ICSE-C '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 485–486.

[29] D. Rodrigues, R. de Melo Pires, E. A. Marconato, C. Areias, J. C. Cunha, K. R. L. J. C. Branco, and M. Vieira, "Service-oriented architectures for a flexible and safe use of unmanned aerial vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 97–109, Spring 2017.